AMENDMENTS TO THE SPECIFICATION

Please amend paragraph 0004 to read:

In addition, implementing "raw" disk I/O over a network can take several seconds. Conventionally, computers expect a relatively fast response from their respective storage devices. If a client (initiator) computer is expecting data I/O to proceed at disk speeds, this network latency can be problematic, since certain network protocols, such as TCP/IP, do not guarantee response times and can take many seconds to complete a single I/O operation.

A1

Please amend paragraph 0006 to read:

Conventionally, small computing devices lack the specialized hardware (such as Fibre Channel hardware) that would enable them to connect to certain types of storage systems, such as a SAN. This ~~lack~~ deficiency is often ~~due to~~ due to the absence of internal "real estate" for additional adapter boards or to the cost-prohibited expense of the Fibre Channel hardware.

A2

Please amend paragraph 0007 to read:

What is needed is a way to allow standard computers to access, ~~and~~ manage, and control storage across a network without needing specialized hardware of dedicated lines, while still maintaining acceptable network latency and access rates. It is also desirable to solve the problem of network latency when storing and retrieving data across a network.

A3

Please amend paragraph 0008 to read:

The present invention operates at least one target storage device over a network. Any computer ("the "client") that can be physically connected to a communications network and that supports some transport protocol over the Internet Protocol (IP) can access block storage devices, such as

A4

disk drives or a Storage Area Network (SAN), via that network connection. The described embodiments of the invention contemplate performing "block" I/O operations, although the invention is not limited only to block data transfers and could also be used for file-based transfers in certain embodiments.

Please amend paragraph 00011 to read:

Although the invention can use any Physical/Data Link layer protocol (~~i.e.~~ e.g., IP, ~~Token Ring, FDDI, etc~~) and any transport layer (~~i.e.~~ e.g., TCP, UDP), one described embodiment uses a TCP/IP protocol run on an Ethernet network. In this embodiment, a computer (the "client") wanting to use storage communicates with another computer (the "target") that "presents" the disks. The target can, in actuality, be a computer, an actual disk drive, or any other system capable of presenting block storage devices to the client over a network connection. Currently DataCore Software Corp. produces such a system called SANSymphony Domain Server (SDS) and the target in at least one described embodiment is an SDS or similar SAN system. Other embodiments use other Physical/Data Link layer protocols (e.g., Token Ring, FDDI, ~~etc.~~) and/or other Transport layers (e.g., UDP).

Please amend paragraph 000112 to read:

A described embodiment has two main software components. One is a device driver that runs on the client (called StpI, where I stands for Initiator). The other is a device driver that runs on the target (called StpT, where T stands for Target). StpI and StpT preferably communicate via a TCP/IP connection. StpI receives commands, such as SCSI commands from the client's device driver, and initiates an appropriate data transfer by communicating with StpT on the target. One command may result in multiple transmissions of data blocks between StpI and StpT.

Please amend paragraph 00013 to read:

A7 There is a third driver, StpC, that runs on the client and implements a cache component. StpC intercepts I/O requests originating from application programs and/or the file system, and determines if requested data can be satisfied from the local cache file that it manages. If not, the commands are passed down to StpI for transmission to the target.

Please amend paragraph 00014 to read:

A8 The commands from StpI are adapted for transmission over the network in accordance with a TCP/IP format, and are transmitted over the network to the target. On the target side, StpT adapts the requests to a format acceptable to the storage device and communicates with the storage device to implement the request.

Please amend paragraph 00025 to read:

A9 The present invention is directed to a novel system for enabling network-capable desktop computers, portable computers and other small computers and devices, preferably having a client application (and collectively referred to herein as "clients," or "client computers"), to access storage devices, such as disk drives and SANs, by using their existing and unmodified network access hardware. In accordance with the present invention, I/O requests embodied as SCSI commands are invoked from a client computer and transported over a network using a standard protocol, such as the TCP/IP protocol, to the storage device.

Please amend paragraph 00027 to read:

A10 In Fig 1(a), a client computer 12 communicates over a network 16 with a server 27, which communicates over a network 16 with a disk drive or disk drives 26.

Please amend paragraph 00028 to read:

A11

In Fig. 1(b), client computer 12 communicates over network 16 with a target device 27², which is attached to a Storage Area Network (SAN) 21.

Please amend paragraph 00031 to read:

A12

The existence of a server application on the target side of a network also allows the client to perform disk administration and to discern storage devices on the target side. A virtual representation of the storage device is presented to the client. In certain embodiments, the virtual storage devices appear as raw disks from the perspective of the client computer, and not as conventional network file systems. Primarily for convenience, the protocol used by the client and target will be referred to collectively as the SCSI Transport Protocol over Internet Protocol (STP/IP). STP/IP allows the client computer to manage the target storage devices as if they were locally connected SCSI disks although they are not, of course~~, not~~.

Please amend paragraph 00034 to read:

A13

Client 112 includes an StpC driver 167 that communicates with a data cache 168 and that generates I/O Request packets (IRPs) 102 if needed; SCSI Disk Driver 162; a SCSI Port Driver 164 that converts the IRPs 102 to SCSI Request Blocks (SRBs) 104; a SCSI miniport driver 166 for converting the SRB²s to SCSI commands 108~~b~~; and an StpI Initiator 169 for transmitting SCSI command frames and WRITE data to the server 122~~,~~ and for listening and receiving READ data from the server 122. The SCSI miniport 166 and StpI socket 169 are referred to collectively as the StpI 115 as indicated by the dotted lines 115. It will be appreciated that the StpI 115 can also include the functions of DCS SCSI Port Driver 164 integrated therein in an alternate embodiment, as indicated by the dashed line 117.

Please amend paragraph 00036 to read:

Cache 168 is preferably stored as files on a local disk of the client computer 112. Thus, cache 168 does not afford high access speeds, as would a conventional memory-based cache, such as a cache used by a file system of the client computer. Because the purpose of the cache 168 is to improve network latency, the described embodiment uses a disk-based cache, which gives slower performance, but allows a much bigger amount of data to be cached (i.e., many gigabits using currently available hard drives). Cache 168 is a block level cache that resides on the initiator, not on the target. A cache "hit" means that the requested data can be found on the local disk drive and, therefore, it is not necessary to spend the relatively long amount of time required to retrieve the data from the target over the network.

A14

Please amend paragraph 00038 to read:

The cache 168 is preferably implemented as a set of files containing the most recently used disk blocks from the target device. The cache 168 is managed by the StpC driver. Data from a SCSI write command is initially passed to the cache 168 and stored. Cache ~~, and cache~~ 168 then forwards the SCSI command to StpI socket module 169 so that the data can also be written to the storage system.

A15

Please amend paragraph 00039 to read:

It will be appreciated by those skilled in the art that, by making extensive use of disk-based caching algorithms, the described embodiment can sometimes avoid having to receive data across the network and can, thus, provide latency results better than if no caching features were ~~are~~ present. Simultaneously writing data to the cache when data is written to the target will ensure that the data is in the cache the next time it is needed.

A16

Please amend paragraph 00045 to read:

A17 If the driver StpC receives a write I/O request 214, StpI writes the data to the target and the data blocks are also written 216 into the cache 168 using a least recently used method. It should be noted that a single read or write may cause the retrieval or storage of multiple data blocks to/from the cache 168.

Please amend paragraph 00052 to read:

A18 The **Frame Sync Number** is assigned by the originator of the frame, usually the initiator. It is a unique number and it is set in two places in the Frame Header, the first four bytes and the last four bytes. Upon receiving a Frame Header, these two values are compared to assure validity of the Frame Header.

Please amend paragraph 00055 to read:

A19 The **Event** field is set before the Frame Header is sent. It indicates the "reason" that this Frame Header is being sent and usually tells the receiver to perform some action on the exchange. It may be one of the following ASCII values.

'NCMD'	-	New SCSI Command is contained in this Frame Header
'SNDD'	-	Send the requested data
'XFER'	-	The data requested follows this Frame Header
'CMPT'	-	The command/exchange is complete
'RSET'	-	Reset

Please amend paragraph 00056 to read:

A20 The **Lun** field contains the Logical Unit Number to which the exchange is directed.

Please amend paragraph 00058 to read:

A21 The **Flags** field is a bit-field ~~the~~ indicating whether data will be sent to the target or received by the initiator.

Please amend paragraph 68 to read:

In summary, the process of StpI 115 interacting with StpT 137 allows SCSI commands originating as disk I/O READ and WRITE commands to be transported over a network using TCP/IP. Use of a block level, disk-based cache on the initiator side of the network allows A22 improvement in network latency and allows the cache to be larger than can generally be achieved with memory-based caches alone. Because the cache operates in parallel and competitively with the main data path, it does not appreciably increase processing time for the transfer of data ~~transfer~~ over the network.